

# **RELATÓRIO DE ATIVIDADES DA IC 19-02-2025**

João Lucas Santos Penha de Oliveira

## **COMUNICAÇÃO**

Com a finalização da comunicação via Bluetooth, é possível comunicar duas ESP32 a partir de dois métodos diferentes: Wifi e Bluetooth. Em continuação com este projeto, os próximos passos foram o desenvolvimento de aplicações que fizesse a comunicação entre as duas BitDogLab a partir de dois ESP32 conectadas via serial com seus respectivos Raspberrys.

Em relação a comunicação serial, a escolhida no momento foi a UART, utilizando as saídas GPIO0 e GPIO1, sendo elas TX e RX respectivamente. Estes pinos são os mesmos utilizados para fazer a comunicação via I2C, podendo ser uma possibilidade futura.

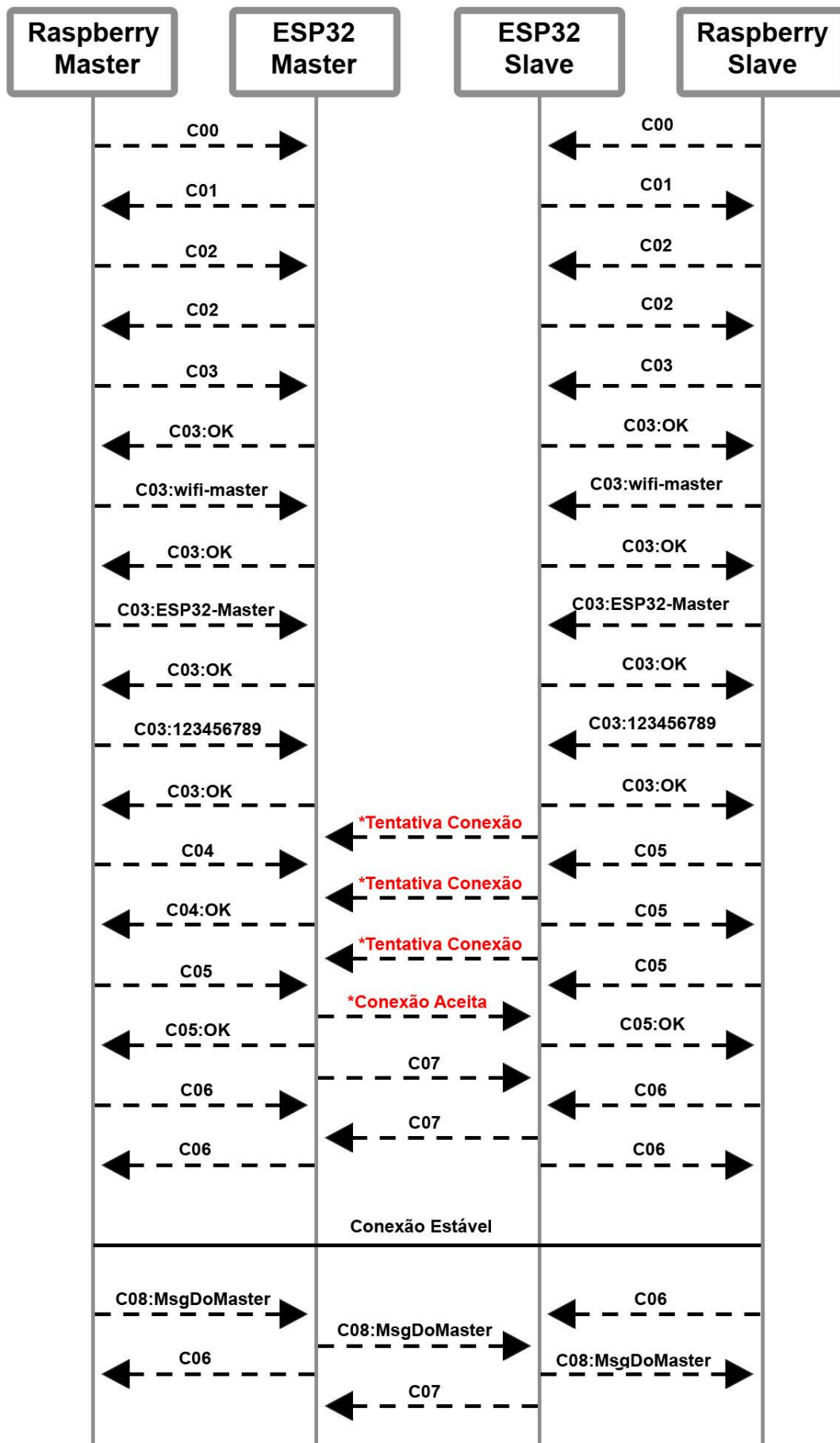
Em relação ao código da comunicação serial, foi adicionado mais robustos ao sistema. Para isso é feito rotinas de verificação a partir de uma máquina de estados, cada estado só pode ser executado após o anterior, até chegar em um estado final em que a comunicação serial foi totalmente estabelecida.

Vale ressaltar que a ESP32 funciona sempre a partir de uma resposta de comunicação da Raspberry, ou seja, a ESP32 só envia uma mensagem serial após receber alguma mensagem anterior, ademais, a mensagem recebida deve estar em um padrão estabelecido. Este padrão escolhido é bem simples, toda mensagem serial deve começar com uma etiqueta referencial a um estado da comunicação, como por exemplo "C00". Desse modo, caso a ESP32 receba essa mensagem via serial, significa que o Raspberry quer estabelecer uma comunicação serial. Perceba que existe uma diferença de comunicação serial quando se trata de uma ESP32 Master ou Slave, pois uma precisa criar a rede e esperar um cliente, enquanto a outra apenas precisa se conectar em uma rede.

Afim de falar dos diversos códigos, abaixo se encontra os códigos tabelados de comunicação:

<b>Código</b>	<b>Master</b>	<b>Slave</b>
C00	Raspberry Tentando se Conectar ao ESP32A	Raspberry Tentando se Conectar ao ESP32A
C01	ESP32A responde ao pedido de conexão	ESP32A responde ao pedido de conexão
C02	Conexão Serial Estabelecida	Conexão Serial Estabelecida
C03*	Envio dos dados da rede a ser criada	Envio dos dados da rede a ser conectado
C04	Aguardando criação da rede	sem uso para slave
C05	Aguardando novo cliente	Aguardando Conexão
C06	Comunicação Padrão Serial	Comunicação Padrão Serial
C07	Comunicação Padrão Rede	Comunicação Padrão Rede
C08*	Nova mensagem	Nova mensagem

Afim de facilitar a explicação de como código age, a figura a seguir ilustra como é estabelecida a conexão.



Como mencionado anteriormente, a ESP32 sempre responde a um pedido da Raspberry. O primeiro pedido é feito mandando o código “C00” para a ESP32, começando então a máquina de estado na ESP32. Neste caso a ESP32 deve responder com o código “C01”. Caso a Raspberry receba esse código, isso significa que está feita a comunicação via serial entre a Raspberry e a ESP32 e caso se limitasse apenas a comunicação serial, seria suficiente mandar apenas “C02” para nunca perder essa conexão.

Um adendo antes de continuar, é que existe timeout na comunicação serial, de modo que, caso a Raspberry ou a ESP32 demore para mandar mensagem, o sistema sempre volta para a primeira etapa da máquina de estados. Por isso é muito importante continuar mandando mensagens sem efeito sempre, para mostrar que a conexão ainda é fiável.

Quando a Raspberry manda o código “C03”, significa que ela quer que a ESP32 trabalhe em algum tipo de rede, no qual vai ser especificado em sua continuação de sua mensagem. No momento funciona da seguinte forma, o Raspberry começa mandando a mensagem “C03” e o ESP32 deve responder com “C03:OK” mostrando que está apto a manipular a rede. Em seguida o Raspberry continua usando o código C03 para enviar os dados da rede, seguindo a seguinte ordem: tipo de rede, nome da rede, senha.

Até o momento, só existem dois tipos de rede, sendo elas: “wifi-master” e “wifi-slave”, o nome da rede e senha pode ser qualquer valor que se queira.

Com os dados da rede enviado, a Raspberry vai aguardar até receber um código “C05:OK”, que representa que ambas ESP32 estão conectadas, e consequentemente as duas Raspberrys. As mensagens devem sempre utilizar do código “C08:[mensagem]”, dessa forma é assegurado que a mensagem chegue no outro dispositivo.

É possível notar que esta biblioteca está versátil e robusta. Versátil, pois a mensagem é transmitida entre os dispositivos independente do que ela esteja escrita, com o limite de 96 bytes (caso queira alterar, basta alterar a constante `BUFFER_SIZE` de todos os códigos). Isso significa que a biblioteca de comunicação pode ser utilizada em qualquer tipo de aplicação, limitada pela latência de comunicação e velocidade de transferência da comunicação sem fio.